AMENDMENTS TO THE CLAIMS

Applicant submits below a complete listing of the current claims, including marked-up claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing. This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of the Claims

1-6. (Canceled)

7. (Currently amended) An object model embodied on a computer-readable medium for managing a service on a computer, the object model comprising:

a policy object model for specifying, by a first user, if it has been determined that the first user is authorized to perform the specification by comparing a rank of the first user against a permitted rank, at least one first policy that the service supports in a packet-centric form, and, by a second user, at least one second policy by selecting a security level from a plurality of security levels, with each security level from the plurality of security levels being previously set for a specified application and a specified user, wherein, when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored, and, when parameters of the application match a condition in an application rule of the policy object model, at least one template is instantiated using at least the stored local port to create at least one policy for the application ~~the policy object model comprises a plurality of policy action classes representing at least a deny, permit and log actions of the service on at least one packet~~; and

a policy engine platform for interacting of the first user with the at least one first policy and of the second user with the at least one second policy, and to provide the at least one first policy and the at least one second policy to at least one component that performs the service, wherein the policy engine platform comprises a rule editor class that is configured to perform at least one of deleting, adding and editing the at least one first policy by the first user, and a setting editor class that is configured to enable the second user to select the security level from the plurality of security levels.

8-10. (Canceled)

11. (Previously presented) The object model of claim 7, wherein the setting editor class is configured to automatically generate a policy based upon an application and user combination.

12. (Previously presented) The object model of claim 11, wherein the setting editor class is configured to generate a plurality of policies, and is further configured to permit said second user to select from the plurality of policies.

13. (Previously presented) The object model of claim 12, wherein the setting editor class is further configured by said second user to permit setting one of the plurality of policies as a default policy.

14. (Previously presented) The object model of claim 7, wherein the policy engine platform comprises a rule explorer for providing a view of the at least one first policy and the at least one second policy.

15. (Previously presented) The object model of claim 7, wherein the policy object model comprises a policyrule object usable to generate a policy, the policyrule object comprising a condition property and an action property, wherein the policy generated by the policyrule object is configured to perform an action specified in the action property responsive to a condition specified in the condition property being met.

16. (Original) The object model of claim 7, wherein the service is a firewall service.

17. (Previously presented) The object model of claim 7, wherein the policy engine platform is configured to deny providing the at least one first policy and/or the at least one second policy to the at least one component if a requestor is not authorized.

18. (Previously presented) The object model of claim 17, wherein determining whether the requestor is authorized comprises comparing a provider rank for the requestor against a permitted provider rank, and if the provider rank for the requestor does not meet or exceed the permitted provider rank, denying the requestor.

19. (Currently amended) A method of managing a service on a computer, the method comprising:

specifying, via a policy object model, by a first user, if it has been determined that the first user is authorized to perform the specification by comparing a rank of the first user against a permitted rank, at least one first policy that the service supports in a packet-centric form, and, by a second user, at least one second policy by selecting a security level from a plurality of security levels, with each security level from the plurality of security levels being previously set for a specified application and a specified user, wherein, when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored, and, when parameters of the application match a condition in an application rule of the policy object model, at least one template is instantiated using at least the stored local port to create at least one policy for the application ~~the policy object model comprises a plurality of policy action classes representing at least a deny, permit and log actions of the service on at least one packet~~;

interacting, via a policy engine platform, of the first user with the at least one first policy, and of the second user with the at least one second policy; and

providing, via the policy engine platform, the at least one first policy and the at least one second policy to at least one component that performs the service, wherein the policy engine platform comprises a rule editor class that is configured to perform at least one of deleting, adding and editing the at least one first policy by the first user, and a setting editor class that is configured to enable the second user to select a security level from the plurality of security levels.

20. (Original) The method of claim 19, further comprising automatically generating a policy based upon an application and user combination.

21. (Previously presented) The method of claim 20, further comprising generating a plurality of policies, and permitting a user to select at least one policy from the plurality of policies.

22. (Previously Presented) The method of claim 21, further comprising setting one of the plurality of policies as a default policy.

23. (Previously presented) The method of claim 22, further comprising authorizing a user prior to allowing the user to select the at least one policy from the plurality of policies.

24. (Currently amended) An object model embodied on a computer-readable medium for managing a firewall service on a computer, the object model comprising:

a policy object model used to specify, by a first user, if it has been determined that the first user is authorized to perform the specification by comparing a rank of the first user against a permitted rank, at least one first policy that the firewall service supports in a packet-centric form, and, by a second user, at least one second policy by selecting a security level from a plurality of security levels, with each security level from the plurality of security levels being previously set for a specified application and a specified user, the policy object model comprising a policyrule object usable to generate a policy, the policyrule object comprising a condition property and an action property, wherein the policy generated by the policyrule object is configured to perform an action specified in the action property responsive to a condition specified in the condition property being met, wherein, when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored, and, when parameters of the application match a condition in an application rule of the policy object model, at least one template is instantiated using at least the stored local port to create at least one policy for the application ~~the policy~~

~~object model comprises a plurality of policy action classes representing at least a deny, permit and log actions of the firewall service on at least one packet~~; and

a policy engine platform comprising a rule editor class that is configured to perform at least one of deleting, adding and editing the at least one first policy by the first user, and a setting editor class that is configured to enable the second user to select a security level from the plurality of security levels.

25. (Original) The object model of claim 24, further comprising an IPSecRule derived from the policyrule object, the IPSecRule being configured to trigger an IPSec callout when an IPSec condition is matched, and to indicate configuration parameters for securing traffic related to the callout.

26. (Original) The object model of claim 25, wherein the IPSecRule evaluates a standard 5-tuple to determine if a condition has been met.

27. (Original) The object model of claim 24, further comprising a KeyingModuleRule derived from the policyrule object, the KeyingModuleRule being configured to select which key negotiation module to use when there is no existing secure channel to a remote peer.

28. (Original) The object model of claim 27, wherein the KeyingModuleRule evaluates a standard 5-tuple to determine if a condition has been met.

29. (Original) The object model of claim 24, further comprising a IKERule derived from the policyrule object and configured to specify the parameters for carrying out Internet Key Exchange key negotiation protocol.

30. (Original) The object model of claim 29, wherein the IKERule evaluates a local address and a remote address to determine if a condition has been met.

31. (Previously presented) The object model of claim 29, wherein the IKERule comprises an IKEAction action property that defines authentication methods for performing Internet Key Exchange key negotiation protocol.